UNIVERSITE CAEN NORMANDIE

Module : Base de données non traditionnelles

Rapport du projet de Visualisation de données

Docker Node.js MongoDB GraphQL D3.js

Réalisé par :

Bassirou THIOUNE 22008769

Mouhamadou Mansour NDJIM 21613317

Enseignant par :

François Rioult

Année Scolaire : 2021/2022

1.Introduction :

Ce présent rapport est le rendu d'un travail portant sur le développement d'une interface web composée de plusieurs vues, et présentant des données de notre choix de la manière la plus accessible possible. Ce projet se place dans la continuité des travaux pratiques initiés en cours. L'objectif est de développer une application permettant à l'utilisateur des interactions avec un thème choisi, sous l'angle de navigation des données. **Docker**, **Node.js, GraphQL, MongoDB et D3.js** sont les outils informatiques de gestion de données et d'applications qui sont utilisés dans le contexte du projet pour en arriver à la réalisation obtenue. L'architecture utilisée dans ce projet se base en partie de l'architecture vu en TP.

2. Les outils : Docker Node.js GraphQL D3.js

2.1. Docker :

Docker est une plate-forme logicielle qui permet de concevoir, tester et déployer des applications rapidement. Il intègre des logiciels dans des unités normalisées appelées conteneurs. Docker nous facilite le déploiement et dimensionnement des applications dans n'importe quel environnement, avec l'assurance que le code s'exécutera correctement.

2.2. Node.js:

Node.js est une plateforme logicielle libre en JavaScript orientée vers les applications hautement concurrentes pouvant monter rapidement en charge. C'est un environnement bas niveau qui permet l'exécution de code JavaScript côté serveur. Il nous permettra de mettre en place assez facilement un serveur web.

2.3. MongoDB:

MongoDB est une base de données NoSQL orientée documents. L'ensemble du système tourne autour de cette gestion de documents, y compris le langage d'interrogation, ce qui en fait son point fort. MongoDB permet de manipuler des objets structurés au format BSON (JSON binaire), sans schéma prédéterminé.

2.4. GraphQL:

GraphQL est un langage de requêtes et un environnement d'exécution, crée par Facebook en 2012, avant d'être publié comme projet open-source en 2015. Il est inscrit dans le modèle Client-Serveur et propose une alternative aux API REST. Il nous permettra de communique et de récupérer des données depuis notre base de données MongoDB.

2.5. D3.js:

D3.js est une bibliothèque JavaScript pour manipuler des documents basés sur des données. Il permet de donner une certaine accessibilité à nos données en utilisant HTML, SVG et CSS. D3.js va nous aider à lier nos données à modèle d'objet de document (DOM), puis d'appliquer des transformations pilotées par les données au document.

3. L'architecture du projet :

L'implémentation de l'application réalisée suit l'architecture suivante :

- Le dossier **data** : qui contient un fichier JSON contenant les données qui seront présentées au niveau de l'interface web (un détail sur ces données sera apporté plus bas).
- Le dossier **mongodata** : qui est un dossier monter lors de la création du conteneur mongo et qui va stocker les fichiers de notre base de données MongoDB en local.
- Le dossier **graphql** : qui contient plusieurs fichiers dont le Dockerfile pour construire le container graphql, le modèle de données graphql (model.graphql) pour la structure des requêtes, ou encore le fichier resolvers.js qui définit les fonctions nécessaires pour la récupération des données.
- Le dossier **ui** : qui contient nos pages html sur lesquelles les données seront présentées (un détail sur ces pages sera apporté plus bas).

4. Description des données :

Les données exploitées dans ce projet sont contenues dans une base de données au format CSV (convertit en JSON via un outil en ligne) de plus 16.000 enregistrements. Ces données résument des informations concernant des jeux vidéo vendus à plus de 100.000 exemplaires un peu partout dans le monde. Le jeu de données en tant que tel a été généré par un scraper de **vgchartz** et on les a directement téléchargées via le site web **Kaggle**.

Cette base de données d'exactement 16.598 enregistrements est composés des champs (clés) suivant :

- "**Rang**" : Classement des ventes globales.
- "**Name**" : Le nom du jeu.
- "**Platform**" : Plate-forme de la sortie des jeux. Exemple : PC, PS4, Xbox, etc.
- "**Year**" : Année de sortie du jeu.
- "**Genre**" : Genre du jeu.
- "**Publisher**" : Editeur du jeu.
- "**NA_Sales**" : Vente en Amérique du Nord (en millions).
- "**EU_Sales**" : Ventes en Europe (en millions).
- "**JP_Sales**" : Vente au Japon (en millions).
- "**Other_Sales**" : Vente dans le reste du monde (en millions).
- "**Global_Sales**" : Total des ventes mondial (en millions).

Notre application a donc pour but de mettre en évidence ces données brutes de manière plus accessible à l'utilisateur, et mettant en exergue certaines informations qui peut-être ne pourraient pas être visible sans un traitement de données au préalable.

Ainsi, à partir des données, on a créé trois agrégations différentes pour les afficher dans trois vues différentes. Ces agrégations sont :

- L'agrégation **publishers** : qui résulte de la variable Publisher. Elle contient la liste de tous les éditeurs de jeu de la base de données, et ainsi pour chaque éditeur, une information que le nombre de jeux qu'il a publiés.

- L'agrégation **plateforms** : qui résulte de la variable Plateform. Elle contient la liste de toutes les plateformes, pour lesquels des jeux ont été développés, et pour chaque plateforme, le nombre de jeux pouvant être joués dessus. Cela nous donne une information sur les plateformes vers lesquelles les développeurs créent plus de jeux.
- L'agrégation **years** : qui résulte de la variable Year. Elle contient pour chaque année entre 1980 et 2020, le nombre de jeux ayant été publiés durant l'année.

5. Les vues :

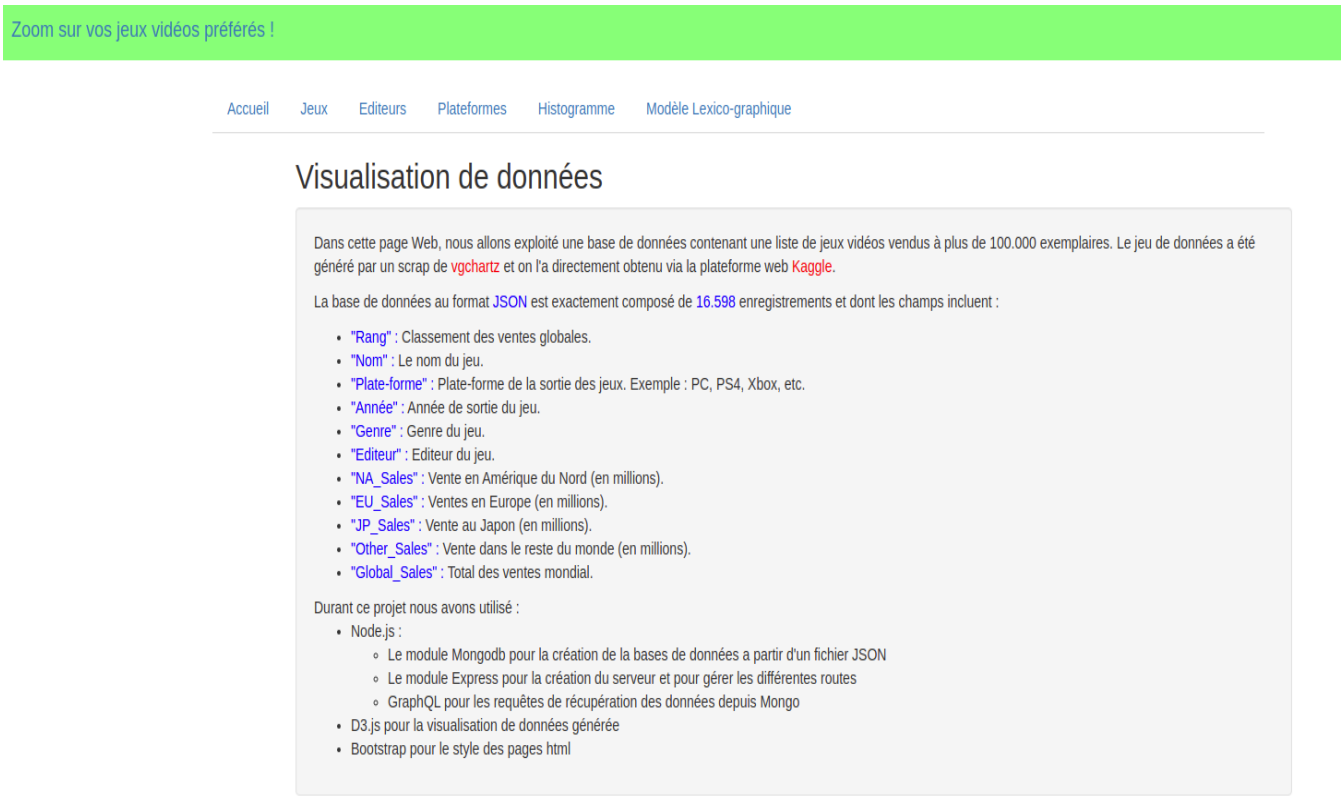
Après avoir récupérer et traiter ces données, il nous faut maintenant les visualiser et interagir avec elles. Pour cela, nous avons proposés une application web avec 6 vues différentes :

- La première vue (index.html) se nomme **Accueil** : elle représente la page d'accueil de l'application. Sur cette page, on présente un résumé des données et de leur provenance.
- La deuxième vue (jeux.html) se nomme **Jeux** : il s'agit d'une page sur laquelle est affichée les informations sur chaque jeu enregistré dans la base de données. Les informations sont disposées dans un tableau dont les colonnes représentent les variables (clés) et les lignes les enregistrements (valeurs).
- La troisième vue (editeurs.html) se nomme **Editeurs** : elle présente les données contenues dans l'agrégation « publishers » de manière plus visuelle. Il s'agit d'un affichage sur deux colonnes avec sur la première le nom de l'éditeur et sur la deuxième le nombre de jeux à son actif. Un bouton de tri est mis à disposition pour faire un trie sur les données par rapport au nombre de publication.
- La quatrième vue (palteformes.html) se nomme **Palteformes** : elle présente les données contenues dans l'agrégation « palteforms » de manière plus visuelle. Il s'agit d'un affichage sur deux colonnes (comme pour la précédente) avec sur la première le nom de la plateforme et sur la deuxième le nombre de jeux publiés pour elle. Un bouton de trie est aussi mis à disposition pour faire un tri sur les données par rapport au nombre de jeux publiés pour cette plateforme.

- La cinquième vue (years.html) se nomme **Histogramme** : elle présente les données contenues dans l'agrégation « years » de manière plus visuelle. Il s'agit d'un affichage par histogramme en barres horizontales des effectifs. On a mis en place un diagramme en bandes renversées où chaque année est représentée par une bande dont largeur est fonction du nombre de jeux publiés durant l'année en question.

- La sixième vue (model.html) se nomme Modèle **Lexico-graphique** : on peut observer sur cette page un nuage de mots représentant les plateformes de jeux les plus appréciées. Plus une plateforme est appréciée et plus sa taille dans le nuage est importante.

6. Aperçu du site :

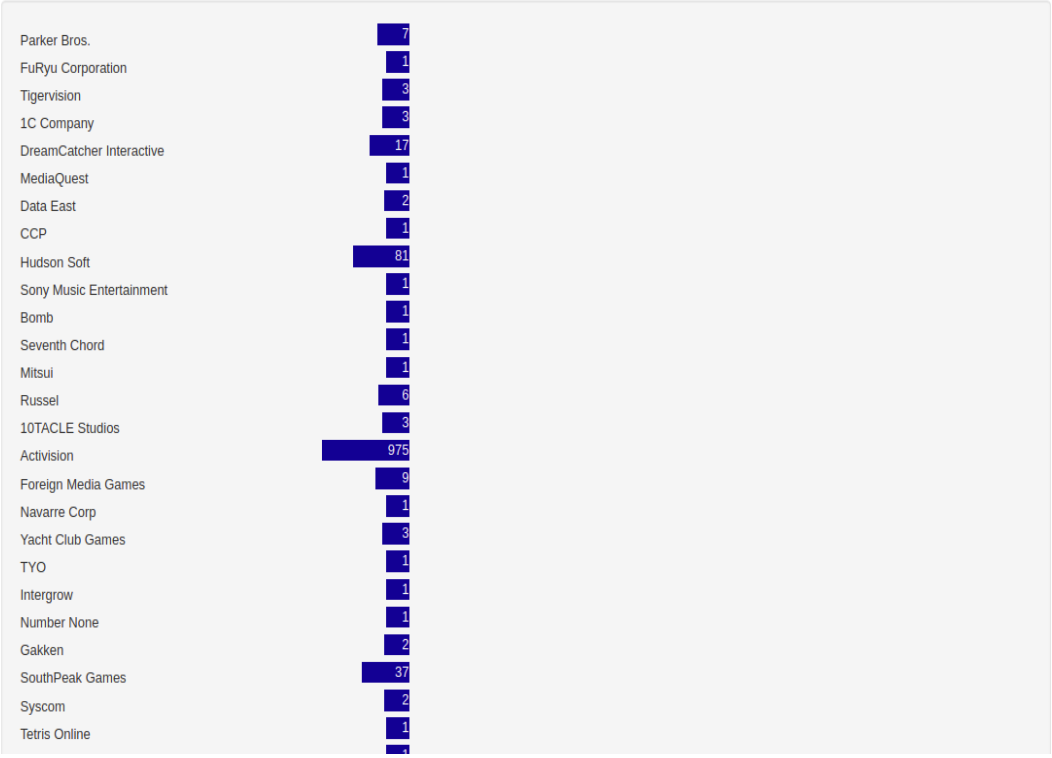


Sur cette page, on peut observer la liste des jeux qui composent notre base de données. On pourra voire pour chaque jeu le nom, la plateforme pour laquelle le jeu a été publié, l'année de publication, le genre, l'éditeur ou encore le bénéfice généré à la vente en millions d'euros.

Nom	Plateforme	Année de publication	Genre	Editeur	Ventes globales(en millions)
Wii Sports	Wii	2006	Sports	Nintendo	82.74
Super Mario Bros.	NES	1985	Platform	Nintendo	40.24
Mario Kart Wii	Wii	2008	Racing	Nintendo	35.82
Wii Sports Resort	Wii	2009	Sports	Nintendo	33
Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	Nintendo	31.37
Tetris	GB	1989	Puzzle	Nintendo	30.26
New Super Mario Bros.	DS	2006	Platform	Nintendo	30.01
Wii Play	Wii	2006	Misc	Nintendo	29.02
New Super Mario Bros. Wii	Wii	2009	Platform	Nintendo	28.62
Duck Hunt	NES	1984	Shooter	Nintendo	28.31
Nintendogs	DS	2005	Simulation	Nintendo	24.76
Mario Kart DS	DS	2005	Racing	Nintendo	23.42
Pokemon Gold/Pokemon Silver	GB	1999	Role-Playing	Nintendo	23.1
Wii Fit	Wii	2007	Sports	Nintendo	22.72
Wii Fit Plus	Wii	2009	Sports	Nintendo	22

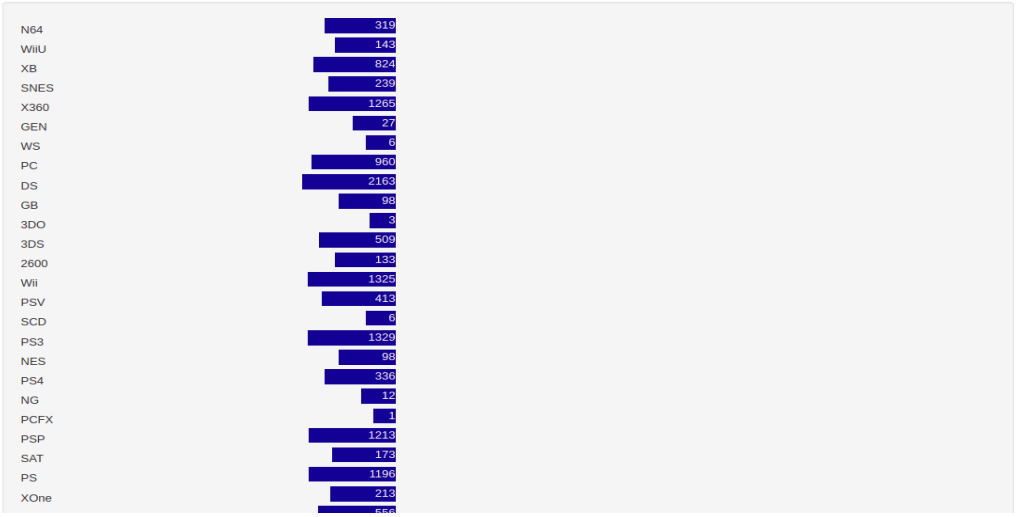
Sur cette page, on peut observer la liste des éditeurs de jeux. Pour chaque éditeur, vous trouverez une information sur le nombre de publications à son actif.

Trie selon le nombre

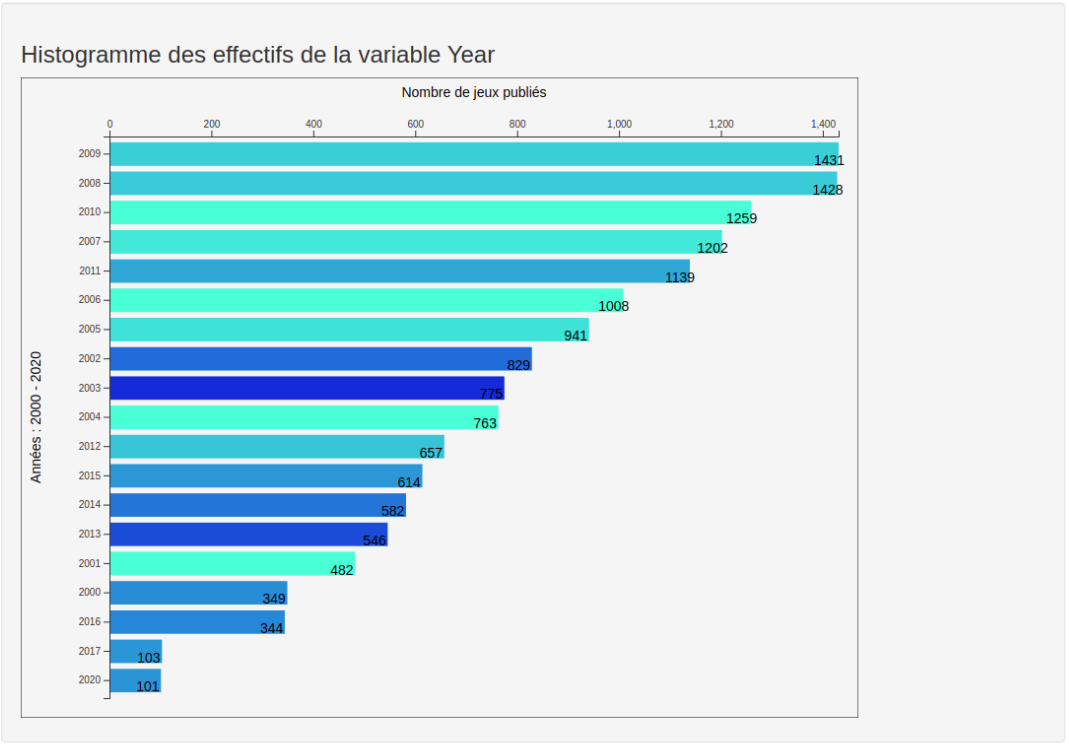


Sur cette page, on peut observer la liste des plateformes sur lesquels les jeux ont été publiés.
Exemple : Xbox, PS3, Wii, etc.
Pour chaque plateforme, vous trouverez une information sur le nombre de jeux qui y sont publiés.

Trie selon le nombre



Ci-dessous, on peut voire un histogramme en barres horizontales des effectifs de la variable Year de notre base de données. Elle représente l'évolution de la publication des jeux entre l'année 2000 et l'année 2020. Chaque barre horizontale de l'histogramme ainsi proposé à une largeur relatif au nombre de jeux publiés durant l'année en ordonnée de la barre.



---

6. Conclusion :

En somme, ce projet nous a permis d'aborder en profondeur les différentes notions apprises en cours et de les mettre en pratique dans un projet applicatif. Il nous a, en outres, permis de comprendre le fonctionnement docker et comment les applications peuvent être déployés beaucoup plus facilement tout en évitant plein de conflits liés à des problèmes de compatibilités au niveau des systèmes.

Références :

[1] https://aws.amazon.com/fr/docker/

[2] https://fr.wikipedia.org/wiki/Node.js

[3] https://openclassrooms.com/fr/courses/4462426-maitrisez-les-bases-de-donnees-nosql/4474601-decouvrez-le-fonctionnement-de-mongodb

[4] https://fr.wikipedia.org/wiki/GraphQL

[5] http://www.vgchartz.com

[6] https://www.kaggle.com/gregorut/videogamesales